**SVILUPPI FUTURI**

Ogni immagine è costituita da un insieme di caratteristiche che sono in grado di rappresentare al meglio il file originale. Queste caratteristiche vengono utilizzate come punto di partenza per parecchi sistemi di visione artificiale. Lo scopo di questo sviluppo futuro è quindi quello di estrarre dall’immagine alcune features che la rendono unica in modo da poterle poi confrontare con un dataset che fa riferimento ad una competenza specifica.

La Key Point Detection può essere fatta tramite i metodi tradizionali, oppure tramite metodi automatizzati. Per quanto riguarda i primi, il riconoscimento dei punti chiave di un’immagine avviene manualmente, cosa che porta ad avere tempi lunghi e bassa precisione. Al contrario, i metodi automatizzati, ossia quelli che rilevano automaticamente i punti, aiutano a migliorare le performance rispetto ai precedenti. A loro volta, questi ultimi hanno due diversi approcci. Il primo viene chiamato *Area Based Matching(ABM)*, si basa sulla registrazione pixel per pixel e segnala la presenza delle funzionalità senza la loro rilevazione. Il secondo è *Feature Based Matching(FBM)*.

Gli elementi caratteristici di un’immagine possono essere angoli, bordi o colori. Anche le caratteristiche si suddividono in due sottogruppi che sono le caratteristiche globali, utilizzate per rappresentare il contenuto dell’immagine, ma hanno lo svantaggio di restituire immagini confuse e quindi hanno un’utilità limitata. Effettuando però operazioni come la segmentazione delle immagini e il campionamento delle stesse, si possono ridurre tali limiti. E le caratteristiche locali per il rilevamento dei key point. Ovviamente, per poter trovare una corrispondenza tra le immagini si devono riconoscere una serie di punti. Per poterli rilevare occorrono quindi dei rilevatori, le cui caratteristiche comprendono la robustezza, la ripetibilità, l’accuratezza, l’efficienza e la quantità.

Un secondo sviluppo da apportare al progetto è quello di realizzare un dataset sintetico basato su una visione a 360° dell’oggetto da rilevare.

Nel progetto attuale la mano viene importata all’interno dell’immagine di background mostrando il palmo alla telecamera oppure con una rotazione massima di 20° per permettere di identificarne anche il profilo.

La visione del dorso non è stata presa in considerazione perché, per forma e dimensioni, è molto simile al palmo e quindi viene rilavato comunque.

Ovviamente andando a creare un dataset contenente anche l’immagine del dorso della mano si riesce ad ottenere una detection, di questa caratteristica, ancora più accurata.

Rimanendo sempre nell’ambito della forma della mano, un’ulteriore miglioramento potrebbe riguardare l’introduzione della detection dei gesti tramite modelli di mano che li riproducono.

Un esempio banale può essere quello dell’identificazione dei numeri andando ad eseguire il training su un dataset sintetico composto da più modelli che riproducono la forma assunta dalla mano per realizzare un determinato valore numerico.

Oppure si potrebbe procedere allo stesso modo ma modificando la tipologia di gesto e realizzare, ad esempio, una sorta di controllore per il gioco “carta–sasso–forbice” che, riconoscendo le tipologie di gesto, va a decretare il vincitore.

In ambito più tecnologico, la detection delle mani e, più precisamente, dei gesti di quest’ultime potrebbe fungere anche da pannello di controllo per varie apparecchiature, ad esempio rilevare un dito verso l’alto potrebbe significare, per un televisore oppure per un forno, rispettivamente “volume su” e “temperatura su”.

Considerando invece, l’approccio con cui si realizza il rilevamento della mano, è possibile sostituire la detection delle mani, tramite video, per puntare sulla detection di oggetti stampati in 3D che rappresentano semplicemente la stampa in tre dimensioni del modello con cui è stata trainata la rete. MOTIVI o ESEMPI??????????????? Dove CAZZO è la collaborazione con Caterina Rizzi????????

Un altro sviluppo, molto simile a quello appena affrontato, consiste nel modificare il dataset sintetico e non l’oggetto da rilevare, ovvero si potrebbe procedere con la scannerizzazione 3D di oggetti reali e si esegue il training utilizzando, come modello, quel preciso scan.

Tramite questo approccio, nel momento in cui si effettua il rilevamento dell’oggetto, si ottiene un risultato eccellente, in termini di accuratezza del detector, ma è possibile imbattersi in un grosso rischio.

Utilizzando come training set il modello scannerizzato di un oggetto è probabile che effettuando la detection su un altro elemento, della stessa categoria di quello usato come modello, quest’ultimo non venga riconosciuto.

Succede questo perché, la scannerizzazione di un oggetto restituisce una rappresentazione 3D molto precisa, quindi effettuando il training su un dataset composto da immagini basate su quel determinato scan, si allena la rete a riconoscere quel particolare oggetto e quindi c’è la probabilità che il detector sia talmente accurato da non riconoscere altri oggetti ad eccezione di quello scannerizzato.

Un ulteriore sviluppo futuro è quello riguardante l’ampliamento del dataset tramite l’inserimento di più modelli di mani 3D diversi tra loro per dimensioni, lunghezza delle dita o colore della pelle. Questa parte è già stata provata durante lo svolgimento del progetto, ma è stato riscontrato un problema durante la fase di training che non permette il corretto funzionamento della rete. PERCHE’???? Quindi si tratterebbe di risolvere il problema legato a tale errore e effettuare l’allenamento, in modo da avere un riconoscimento delle mani generalizzato e non fortemente influenzato dall’unico modello presente.

Infine, durante il lavoro, è stata utilizzata una versione non aggiornata di TensorFlow, ovvero la 1.15. Sarebbe opportuno migrare il tutto all’ultima versione 2.0 perché sono state rimosse le API ridondanti e inoltre si integra meglio con il runtime di Python. La migrazione è già stata provata durante la realizzazione del progetto riscontrando però un problema legato a ssd\_mobilenet\_v2 che non veniva supportato dalla versione TensorFlow utilizzata.